



CODER HOUSE

PROYECTO FINAL

SQL FLEX

ENTREGA FINAL

Autora: Ing. Samira Godoy Mir

1. Descripción de la temática de la base de datos

1.1. Introducción

La base de datos está diseñada para gestionar el proceso logístico de una empresa de transporte. Permite registrar clientes, choferes, vehículos, pedidos, viajes, facturas y mantenimiento de vehículos. El objetivo es optimizar la gestión de envíos, el control financiero y la administración de la flota de transporte.

Esta base de datos servirá como el núcleo de un sistema que centraliza la información logística de una empresa de transporte, brindando trazabilidad y eficiencia en la administración de los recursos.

1.2. Objetivos

Objetivo general:

- Diseñar e implementar una base de datos para mejorar la gestión logística de una empresa de transporte.

Objetivos específicos:

- Almacenar información de clientes, choferes y vehículos de manera estructurada.
- Registrar y gestionar pedidos, asegurando su trazabilidad en todo el proceso de transporte.
- Administrar los viajes con información detallada sobre origen, destino, fecha y vehículo asignado.
- Controlar la facturación y el estado de los pagos de los servicios prestados.
- Llevar un registro de mantenimiento de los vehículos para evitar problemas operativos.

1.3. Situación problemática

Las empresas de transporte que no cuentan con un sistema de gestión adecuado suelen enfrentarse a problemas como:

- **Falta de control sobre los pedidos y entregas**, lo que genera retrasos y errores en las entregas.
- **Dificultades en la gestión de facturación y cobros**, afectando el flujo de ingresos.
- **Desorden en la asignación de vehículos y choferes**, causando ineficiencias en los viajes.
- **Desconocimiento del estado de mantenimiento de los vehículos**, lo que incrementa el riesgo de fallas y costos de reparación inesperados.

La implementación de una base de datos organizada y optimizada permitirá solucionar estos inconvenientes, mejorando la administración de los recursos y la calidad del servicio.

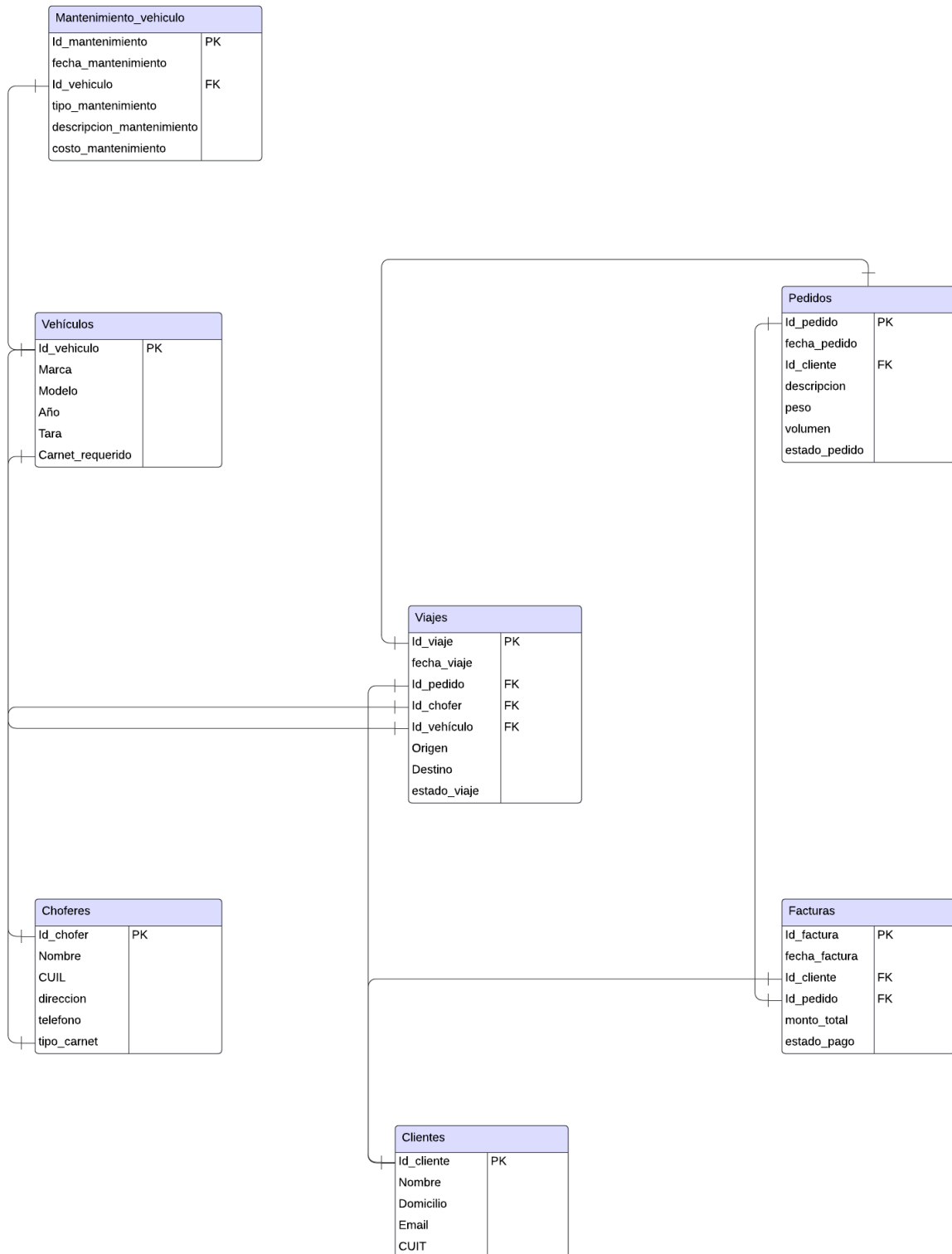
1.4. Modelo de negocio

La empresa opera bajo un modelo B2B (Business to Business), prestando servicios de transporte a otras empresas que requieren trasladar mercadería de un punto a otro de manera eficiente y segura.

Principales actores del modelo:

- **Clientes:** Empresas que contratan el servicio de transporte.
- **Choferes:** Empleados responsables de realizar los viajes.
- **Flota de vehículos:** Camiones o furgonetas utilizadas para la distribución de los pedidos.

2. Diagramas de Entidad-Relación (DER)



3. Listado de tablas

Clientes			
Almacena la información de los clientes de la empresa.			
Nombre del campo	Descripción	Tipo de datos	Tipo de clave
id_cliente	Identificador único del cliente.	INT, AutoIncrement	PK
nombre	Nombre del cliente.	VARCHAR(100), NOT NULL	
CUIT	Identificación fiscal.	VARCHAR(20), UNIQUE, NOT NULL	
direccion	Dirección del cliente.	VARCHAR(100), NOT NULL	
email	Correo electrónico del cliente.	VARCHAR(100)	

Choferes			
Almacena información sobre los choferes de la empresa.			
Nombre del campo	Descripción	Tipo de datos	Tipo de clave
id_chofer	Identificador único del chofer.	INT, AutoIncrement	PK
nombre	Nombre del chofer.	VARCHAR(100), NOT NULL	
CUIL	Identificación fiscal del chofer.	VARCHAR(20), UNIQUE, NOT NULL	

direccion	Dirección del chofer.	VARCHAR(100), NOT NULL	
telefono	Teléfono del chofer.	VARCHAR(20), NOT NULL	
tipo_carnet	Tipo de carnet de conducir.	ENUM('A', 'B', 'C', 'D', 'E', 'F', 'G'), NOT NULL	

Vehículos			
<i>Registra los vehículos utilizados en los viajes.</i>			
Nombre del campo	Descripción	Tipo de datos	Tipo de clave
id_vehiculo	Identificador único del vehículo.	INT, AutoIncrement	PK
marca	Marca del vehículo.	VARCHAR(100), NOT NULL	
modelo	Modelo del vehículo.	VARCHAR(100), UNIQUE, NOT NULL	
año	Año de fabricación del vehículo.	INT, NOT NULL	
tara	Peso del vehículo vacío, en toneladas.	DECIMAL(10, 2) NOT NULL	
carnet_requerido	Tipo de carnet necesario para conducir el vehículo.	ENUM('A', 'B', 'C', 'D', 'E', 'F', 'G'), NOT NULL	

peso_maximo	Capacidad máxima de carga de cada vehículo.	DECIMAL(10,2) NOT NULL	
-------------	---	---------------------------	--

Pedidos			
<i>Contiene los pedidos realizados por los clientes.</i>			
Nombre del campo	Descripción	Tipo de datos	Tipo de clave
id_pedido	Identificador único del pedido.	INT, AutoIncrement	PK
fecha_pedido	Fecha de creación del pedido.	TIMESTAMP, DEFAULT CURRENT_TIME STAMP	
id_cliente	Cliente que realiza el pedido.	INT, NOT NULL	FK
descripcion_pedido	Descripción del pedido.	VARCHAR(500), NOT NULL	
peso	Peso del pedido en kg.	(DECIMAL(10,2), NOT NULL)	
volumen	Volumen del pedido en m³	(DECIMAL(10,2), NOT NULL)	
estado_pedido	Estado del pedido.	ENUM, DEFAULT 'Pendiente'	

Viajes			
<i>Registra los viajes que transportan pedidos.</i>			
Nombre del campo	Descripción	Tipo de datos	Tipo de clave
id_viaje	Identificador único del viaje.	INT, AutoIncrement	PK
fecha_viaje	Fecha y hora del viaje.	TIMESTAMP, DEFAULT CURRENT_TIMESTAMP	
id_pedido	Pedido asociado al viaje.	INT, NOT NULL	FK
id_vehiculo	Vehículo utilizado en el viaje.	INT, NOT NULL	FK
id_chofer	Chofer que realizó el viaje.	INT, NOT NULL	FK
origen	Lugar de origen del viaje.	VARCHAR(255), NOT NULL	
destino	Lugar de destino del viaje.	VARCHAR(255), NOT NULL	
estado_viaje	Estado del viaje.	ENUM, DEFAULT 'Pendiente'	

Facturas			
Registra las facturas generadas.			
Nombre del campo	Descripción	Tipo de datos	Tipo de clave
id_factura	Identificador único de la factura.	INT	PK
fecha_factura	Fecha de emisión de la factura.	TIMESTAMP, DEFAULT CURRENT_TIME STAMP	
id_cliente	Cliente al que se factura	INT, NOT NULL	FK
id_pedido	Pedido facturado	INT, NOT NULL	FK
monto_factura	Monto total de la factura.	DECIMAL(10,2), NOT NULL	
estado_pago	Estado del pago	ENUM, DEFAULT 'Pendiente'	

Mantenimiento_vehiculo			
Registra los mantenimientos de los vehículos.			
Nombre del campo	Descripción	Tipo de datos	Tipo de clave

id_mantenimiento	Identificador único del mantenimiento	INT, AutoIncrement	PK
fecha_mantenimiento	Fecha del mantenimiento.	TIMESTAMP, DEFAULT CURRENT_TIMESTAMP	
id_vehiculo	Vehículo al que se le realiza mantenimiento.	INT, NOT NULL	FK
tipo_mantenimiento	Tipo de mantenimiento: Preventivo o correctivo	ENUM, NOT NULL	
descripcion_mantenimiento	Descripción del mantenimiento.	TEXT, NOT NULL	
costo_mantenimiento	Costo del mantenimiento.	DECIMAL(10,2)	

4. Vistas

4.1 Vista N°1: Viajes completados por chofer.

Descripción:	Muestra la cantidad total de viajes completados por cada chofer, ordenados de mayor a menor.
Objetivos:	<p>Analizar la cantidad de viajes exitosamente finalizados por cada chofer.</p> <p>Evaluar el desempeño de los choferes en función de la cantidad de viajes completados.</p> <p>Identificar choferes con mayor o menor actividad.</p>
Tablas involucradas:	<p>choferes (c): Contiene los datos de los choferes.</p> <p>viajes (v): Contiene los viajes realizados y su estado.</p>

4.2 Vista N°2: Ingresos por cliente.

Descripción:	Muestra el total facturado por cada cliente, considerando únicamente facturas pagadas.
Objetivos:	<p>Identificar clientes que generan mayores ingresos.</p> <p>Evaluar la relación comercial con cada cliente.</p> <p>Facilitar análisis financieros sobre la facturación total.</p>
Tablas involucradas:	<p>clientes (cl): Contiene los datos de los clientes.</p> <p>facturas (f): Contiene los montos facturados y su estado de pago.</p>

4.3 Vista N°3: Pedidos pendientes de entrega.

Descripción:	Lista de pedidos que aún no han sido entregados ni cancelados.
Objetivos:	Identificar pedidos en curso o con demoras. Gestionar y optimizar la logística de entregas. Priorizar la resolución de pedidos pendientes.
Tablas involucradas:	pedidos (p): Contiene los pedidos y su estado. clientes (cl): Contiene la información de los clientes.

4.4 Vista N°4: Resumen de viajes y carga transportada.

Descripción:	Muestra cuántos viajes se realizaron y cuánta carga se transportó en total.
Objetivos:	Evaluar la eficiencia del transporte y la logística. Analizar la cantidad de carga movida en los viajes. Optimizar la planificación del transporte según la demanda.
Tablas involucradas:	viajes: Contiene los datos de los viajes realizados. pedidos: Contiene el peso de los pedidos transportados en cada viaje.

4.5 Vista N°5: Facturas pendientes de cobro.

Descripción:	Lista de facturas que aún no han sido pagadas, agrupadas por cliente.
Objetivos:	Identificar clientes con pagos pendientes. Evaluar el monto total de deuda de cada cliente. Ayudar en la gestión de cobranzas y control financiero.
Tablas involucradas:	facturas (f): Contiene los datos de las facturas y sus estados de pago.

5. Funciones

5.1 Función N°1: pedido_entregado(id_pedido INT)

Descripción:	Esta función verifica si un pedido específico ha sido entregado. Si el pedido tiene el estado "Entregado", devuelve 1 (verdadero); en caso contrario, devuelve 0 (falso).
Objetivos:	<p>Facilitar la validación del estado de entrega de un pedido.</p> <p>Poder utilizar la función en consultas o procedimientos almacenados que requieran confirmar si un pedido ya fue entregado.</p> <p>Optimizar el control de la logística y el seguimiento de pedidos.</p>
Tablas involucradas:	pedidos: Contiene los pedidos registrados con su estado actual.

5.2 Función N°2: calcular_peso_total_viaje(id_viaje INT)

Descripción:	Esta función calcula el peso total transportado en un viaje específico, sumando los pesos de todos los pedidos asignados a ese viaje.
Objetivos:	<p>Determinar cuánto peso transporta un viaje en función de los pedidos asignados.</p> <p>Ayudar a controlar que el peso total del viaje no exceda la capacidad del vehículo.</p> <p>Facilitar cálculos logísticos y optimización del transporte.</p>
Tablas involucradas:	<p>pedidos: Contiene los datos de los pedidos, incluyendo su peso.</p> <p>viajes: Relaciona los pedidos con un viaje específico.</p>

6. Stored Procedures

6.1 Stored procedure N°1: actualizar_estado_viaje

Descripción:	Este procedimiento actualiza el estado de un viaje y guarda un historial de los cambios de estado. Antes de actualizar, consulta el estado actual del viaje y, después de la actualización, almacena el cambio en la tabla historial_estados_viajes.
Objetivos:	<p>Mantener un registro histórico de los cambios de estado de los viajes.</p> <p>Mejorar la trazabilidad del estado de cada viaje.</p> <p>Permitir auditorías y seguimiento de retrasos o cancelaciones.</p> <p>Facilitar la gestión logística con un historial de cada viaje.</p>
Tablas involucradas:	<p>viajes: Contiene la información de cada viaje.</p> <p>historial_estados_viajes: Guarda los cambios de estado con la fecha y hora exacta.</p>

6.2 Stored procedure N°2: asignar_pedidos_a_viaje

Descripción:	Este procedimiento asigna pedidos a un viaje sin exceder la capacidad del vehículo. Recorre los pedidos disponibles (ordenados por peso ascendente) e intenta agregarlos hasta que se alcance el límite de peso del vehículo asignado al viaje.
Objetivos:	<p>Optimización del uso de los vehículos asegurando que se carguen hasta su capacidad máxima sin sobrepasarla.</p> <p>Automatización de la asignación de pedidos a viajes, evitando errores humanos.</p> <p>Reducción de costos logísticos al maximizar el uso de los vehículos y minimizar viajes con poca carga.</p>

Tablas involucradas:	<p>viajes: Contiene la información del viaje al cual se asignarán los pedidos.</p> <p>vehiculos: Se consulta la capacidad máxima de carga del vehículo.</p> <p>pedidos: Se obtienen los pedidos disponibles para asignar al viaje.</p> <p>viajes_pedidos (tabla intermedia): Relaciona pedidos con viajes y almacena su peso y volumen.</p>
----------------------	---

7. Triggers

7.1 Trigger N°1: actualizar_estado_viaje

Descripción:	El trigger actualizar_estado_viaje se activa automáticamente después de que se actualiza un registro en la tabla pedidos. Su objetivo es verificar si todos los pedidos asociados a un viaje han sido entregados y, si es así, actualizar el estado del viaje a "Completado".
Objetivos:	Garantizar que un viaje solo se marque como "Completado" cuando todos los pedidos que lo conforman tengan estado "Entregado". Mantener la integridad de la base de datos evitando actualizaciones manuales en la tabla viajes.
Tablas involucradas:	pedidos → Contiene los pedidos con su estado actual. viajes_pedidos → Tabla intermedia que relaciona pedidos con viajes. viajes → Registra los viajes y su estado (Pendiente, En tránsito, Completado, Cancelado).
Situaciones en las que se acciona:	Si todos los pedidos del viaje están entregados, el estado del viaje pasa a "Completado". Si queda al menos un pedido pendiente, el estado del viaje no cambia.

7.2 Trigger N°2: verificar_disponibilidad_chofer

Descripción:	Este trigger se ejecuta antes de actualizar la tabla viajes. Si se intenta asignar un chofer a un viaje cuando ya tiene otro en tránsito, el trigger bloquea la actualización y genera un error.
Objetivos:	Evitar que un chofer tenga dos viajes en tránsito al mismo tiempo. Asegurar una correcta planificación logística al impedir asignaciones erróneas. Mejorar la eficiencia en la gestión de choferes, asegurando que no se sobrecarguen de trabajo.

Tablas involucradas:	viajes: Se verifica que el chofer no tenga otro viaje en tránsito antes de asignarle uno nuevo.
Situaciones en las que se acciona:	Si el trigger funciona, debe mostrar un error como: <i>"Error: No se puede eliminar un pedido ya entregado".</i>

8. Archivo SQL

8.1 Script creación de objetos.

A través del siguiente link se puede acceder al script:

https://github.com/Samirarg/Coderhouse_SQLFlex

8.2 Script de inserción de datos en las bases.

En el repositorio: https://github.com/Samirarg/Coderhouse_SQLFlex se encuentran también los archivos CSV que deben cargarse en el siguiente orden con el Table Data Import Wizard:

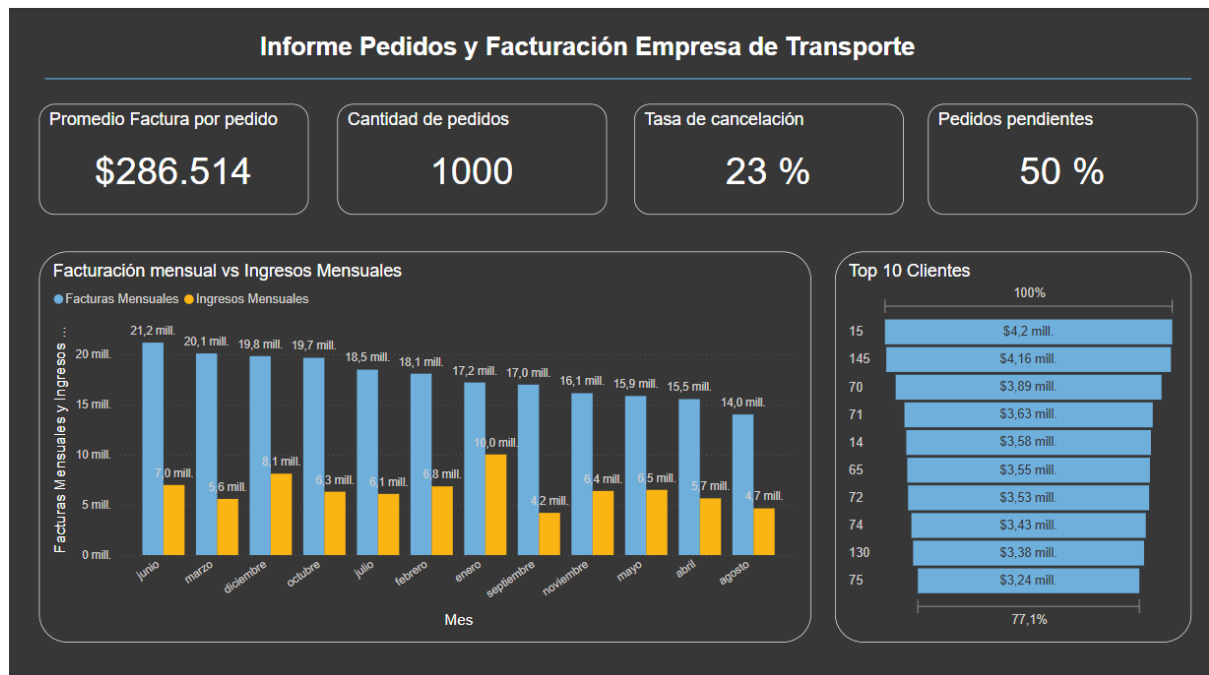
- 1) TablaClientes.csv
- 2) TablaChoferes.csv
- 3) TablaVehiculos.csv
- 4) TablaFacturas.csv
- 5) TablaPedidos.csv
- 6) TablaViajes.csv
- 7) TablaMantenimiento.csv

8.3 Script de creación de Vistas, Funciones, Stored Procedures y Triggers.

A través del siguiente link se puede acceder al script:

https://github.com/Samirarg/Coderhouse_SQLFlex/blob/main/Proyecto_GodoyMir_2Entrega.sql

9. Informes



La imagen anterior presenta una captura del informe desarrollado en Power BI a partir del dataset de una empresa de transporte ficticia, utilizado para el proyecto. Este informe permite una rápida visualización y análisis de indicadores clave del negocio.

Se destacan tarjetas visuales que muestran:

- El valor promedio de factura por pedido.
- La cantidad total de pedidos realizados.
- La tasa de cancelación, que refleja el porcentaje de pedidos cancelados respecto al total.
- La tasa de pedidos pendientes, representando aquellos aún no entregados o procesados.

Además, se incluye un gráfico de columnas agrupadas que compara el monto total facturado con el total efectivamente pagado. Finalmente, un gráfico de embudo muestra el top 10 de clientes según su volumen de facturación, facilitando la identificación de los principales generadores de ingresos.

10. Herramientas y tecnologías utilizadas

En el desarrollo del presente proyecto se utilizaron diversas herramientas digitales que facilitaron tanto la construcción como la documentación del sistema. Para la generación de los datasets ficticios se utilizó la página **Mockaroo**, que permite crear datos simulados con estructuras realistas. El modelo entidad-relación (E-R) fue diseñado con **Lucidchart**, una herramienta visual que posibilita representar gráficamente las tablas y sus relaciones. La base de datos fue implementada y gestionada mediante **MySQL Workbench 8.0 CE**, utilizada para crear estructuras, ejecutar consultas SQL y realizar pruebas. Para la visualización de los resultados y análisis de los datos, se utilizó **Power BI**, que permitió generar informes interactivos y dinámicos.

Además, se utilizó **GitHub** como repositorio para compartir y versionar los archivos del proyecto, incluyendo los datasets en formato .csv y los scripts en .sql. Por último, las entregas y documentación del proyecto fueron redactadas y organizadas a través de **Google Docs**, facilitando el trabajo ordenado y colaborativo.